ANÁLISIS COMPARATIVO DE METAHEURÍSTICAS APLICADAS AL PROBLEMA DEL TSP.

Juan Adolfo Montesino Guerra[a], Héctor José Puga Soberanes[a], Manuel Ornelas Rodríguez[a], Juan Martín Carpio Valadez[a] y Uriel Ervey Bernal Magallanes[a]

[a]Instituto Tecnológico de León, León, Gto. adolfo.montesino@hotmail.com, pugahector@yahoo.com, mornelas67@yahoo.com.mx, jmcarpio61@hotmail.com, ing.uriel.bernal@gmail.com

RESUMEN

El problema del agente viajero (TSP por sus siglas en ingles), es un problema combinatorio conocido, resuelto y abierto a investigación por su complejidad de resolución. Este problema es importante porque se puede relacionar con muchos otros problemas, tales como el diseño de rutas de transporte o como el ciclo que debe de seguir un taladro de placas de circuito impreso, entre otros. Estos se han solucionado, entre otras, por medio de técnicas metaheurísticas.

Las metaheurísticas son estructuras algorítmicas generales adaptables a muchos problemas de optimización. Muchos de estos métodos han sido ampliamente estudiados en las últimas décadas, resultando en una gran variedad de algoritmos heurísticos con buenos rendimientos en la resolución de problemas, y que a menudo superan incluso heurísticas especializadas.

Metodologías como Algoritmo Genético (AG), Algoritmo Memético (AM) y Sistema de Hormigas (AS), se han aplicado en nuestra investigación mostrando resultados aceptables y de rendimiento favorable de acuerdo al consumo de recursos computacionales. En este trabajo además se presenta un análisis comparativo, haciendo uso de estadísticas no paramétricas, con Algoritmo de Sistema Inmune (ASI) y Optimización por Colonia de Hormigas (ACO) resolviendo el TSP. Esto permitirá establecer las bases para una investigación posterior del enfoque hiperheurístico para resolver el TSP.

1. INTRODUCCIÓN

Existen problemas de optimización combinatoria complejos en diversos campos como la economía, el comercio, la ingeniería, la industria o la medicina. Sin embargo, a menudo estos problemas son muy difíciles de resolver en la práctica. El estudio de esta dificultad inherente para resolver dichos problemas tiene cabida en el campo de la teoría de las Ciencias de la Computación, ya que muchos de ellos pertenecen a la clase de problemas NP-duros, lo que significa que no existe un algoritmo conocido que los resuelva en un tiempo polinomial [1].

Día tras día, siguen apareciendo nuevos problemas de este tipo, lo que ha dado lugar a que se hayan realizado muchas propuestas de algoritmos para tratar de solucionarlos. Las técnicas existentes se pueden clasificar básicamente en algoritmos exactos o aproximados. Los algoritmos exactos intentan encontrar una solución optima y demostrar que la solución obtenida es de hecho la óptima global, la cual consiste en encontrar el extremo global (máximo o mínimo) de una función matemática (función objetivo) en alguna región de interés [2]. Los algoritmos aproximados se pueden clasificar en dos tipos principales: algoritmos constructivos y algoritmos de búsqueda local.

Los primeros se basan en generar soluciones desde cero, añadiendo componentes (elementos) hasta completar una solución iterativamente [1] y los algoritmos de búsqueda local están basado en el mejoramiento paso a paso de la función de costo al explorar las vecindades de soluciones cercanas, contrario a los métodos exactos, que garantizan dar una solución óptima al problema [3].

Las metaheurísticas o algoritmos aproximados incorporan conceptos de muchos y diversos campos como la genética, la biología, la inteligencia artificial, las matemáticas, la física y la neurología, entre otras. Algunas de estas metaheurísticas son AS, ACO, AG, AM y SI [1].

2. TEORÍA

2.1 Travelling salesman problem

El problema del agente viajero (TSP por sus siglas en ingles) se remonta a la Europa del siglo XVII, donde era común contar con un agente viajero que llevaba un catálogo con productos para atender la demanda de ciudades, pueblos y aldeas alejadas de las ciudades principales. En 1800 el matemático irlandés William Rowand Hamilton diseñó un juego para dos competidores, el juego consiste en hacer un recorrido por 20 puntos de un icosaedro usando las conexiones de la figura e iniciando y terminando en el mismo punto. La conexión entre el juego de Hamilton y el TSP consiste en que en los dos se busca una ruta que inicie y termine en el mismo lugar sin repetir puntos visitados.

Este problema es uno de los más estudiados en la optimización combinatoria NPduro [4], [5]. En la literatura existe un gran número de aproximaciones para resolverlo tanto con técnicas exactas como con técnicas heurísticas y Metaheurísticas. Formalmente la propuesta por Dantzig, Fulkerson y Johnson [*6*] es una de las más aceptadas por la comunidad académica y consiste en:

Dado un conjunto de n ciudades $V = \{1, 2, 3, ..., n\}$, y un conjunto de arcos $(i, j) \in A$ uniendo cada una de las ciudades, donde A es el espacio de búsqueda. Si C_{ij} es la distancia para ir de la ciudad i a la ciudad j donde $(\mathrm{C}_{ij} = \mathrm{C}_{ji})$ en el caso simétrico y X_{ij} la variable de decisión del problema es:

$$x_{ij} = \begin{cases} 1 \text{ Si el arco (i,j) es usado para hacer el tour} \\ 0 \text{ En caso contrario} \end{cases}$$

El modelo matemático asume la siguiente forma:

$$\min Z(\mathrm{x}) = \sum_{(i,j)\in A} c_{ij} x_{ij} \qquad (1.1)$$

$$\sum_{\{\mathrm{i}:\ (\mathrm{i,j})\in \mathrm{A}\}} x_{ij} = 1 \qquad \forall j \in V \qquad (1.2)$$

$$\sum_{\{\mathrm{j}:\ (\mathrm{i,j})\in \mathrm{A}\}} x_{ij} = 1 \qquad \forall i \in V \qquad (1.3)$$

$$\sum_{\{(i,j)\in A:\ i\in U,\ j\in (V-U)\}} x_{ij} = 1 \qquad 2 \le |U| \le |V| - 2 \tag{1.4}$$

Donde A es el espacio de búsqueda, c_{ij} es el peso de la arista o la distancia asociada a x_{ij}, la ecuación (1.1) corresponde al cálculo de la función objetivo. La restricción (1.2) indica que se puede llegar a cada ciudad desde una única ciudad anterior. La restricción (1.3) indica que desde la ciudad *i* se puede pasar a una única ciudad (de la ciudad *i* se puede salir por un único camino). La restricción (1.4) evita que se generen subtours.

El problema tiene evidentes aplicaciones prácticas en las áreas de logística de transporte que cualquier negocio de reparto, pequeño o grande, conoce. Pero tiene otras aplicaciones no tan evidentes como en robótica, un ejemplo es que permite resolver problemas de fabricación para minimizar el número de desplazamientos al realizar una serie de perforaciones en una plancha o en un circuito impreso. También puede ser utilizado en la generación de rutas de vehículos, manufactura flexible, diseño de redes de computadoras entre otros.

2.2 Algoritmo Genético

Los algoritmos genéticos forman parte de la Computación Evolutiva, que constituye una familia de modelos computacionales inspirados en la evolución natural [7]. Los algoritmos gen éticos constituyen el paradigma más completo de los que presenta la Computación Evolutiva. Una característica importante es el poco conocimiento especıfico que precisan, del problema al que se aplican, para su funcionamiento. Permiten resolver problemas con poco esfuerzo computacional, especialmente en los casos en que otros métodos fallan o suponen unos requerimientos computacionales excesivos [7].

Utilizan una población de individuos, la cual evoluciona al ser sometidos dichos individuos a una serie de transformaciones mediante unos determinados operadores. Se emulan los procesos de selección natural y de reproducción presentes en la Naturaleza, siendo los individuos más fuertes los que sobreviven y procrean a lo largo de la ejecución del algoritmo [7].

Las etapas del algoritmo genético, que denotan el proceso interno son Elitismo, Selección, Cruza y Mutación.

2.3 Algoritmo Memético

Los algoritmos meméticos son técnicas de optimización que combinan sinérgicamente conceptos tomados de otras metaheurísticas, tales como la búsqueda basada en poblaciones (como en los algoritmos evolutivos), y la mejora local (como en las técnicas de seguimiento del gradiente) [8]. Los orígenes de los algoritmos memético se remontan a finales de los años ochenta. En aquella época, el campo de la computación evolutiva estaba comenzando a afianzarse sólidamente. Fue cuando surgió la idea básica que sustenta a los algoritmos memético: combinar conceptos y estrategias de diferentes metaheurísticas, para intentar anular las desventajas de las mismas [8].

La idea central de los algoritmos memético es: mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competiciones [8].

2.4 Algoritmo Sistema Inmune

El sistema inmune artificial es un modelo computacional de nuestro sistema inmune biológico que tiene la capacidad de realizar algunas tareas como el aprendizaje, adquisición de memoria, generación de diversidad, tolerancia al ruido, generalización, detección distribuida y optimización; basados en los principios inmunológicos, nuevas técnicas computacionales han sido desarrolladas enfocándose no solo a una mejor comprensión del sistema mismo, sino también a la solución de problemas de ingeniería [9].

2.5 Sistema de Hormigas (Ant System)

El algoritmo AS fue introducido por Dorigo en 1992 [10]. Es un algoritmo de búsqueda inspirado por el comportamiento de hormigas reales. Donde cada hormiga construye una ruta desde el hormiguero hasta la comida siguiendo estocásticamente (aleatorio) cantidades de niveles de feromonas, la intensidad de las feromonas situadas sesgará la trayectoria de hormigas sucesoras.

El algoritmo de Hormigas (AH) fue desarrollado como una heurística de propósito general que puede ser usado para solucionar diferentes problemas de optimización combinatoria. Esta heurística tiene las características de ser versátil, robusta y tener un enfoque basado en poblaciones. [10]

2.6 Optimización por colonia de hormigas (Ant colony Optimization)

El algoritmo ACO está inspirado en el comportamiento real de las hormigas. Estos insectos son capaces de encontrar la ruta más corta entre su colonia y una fuente de alimento. Esto se debe a que las hormigas pueden “transmitir” información entre ellas, gracias a un rastro de feromona que cada una de ellas deja al desplazarse. Cuando una hormiga descubre una fuente de alimento retorna a la colonia, siguiendo el rastro de feromona y reforzando el depósito de esta. La concentración de mayor de esta sustancia en este camino atrae a otras hormigas de la colonia, las cuales en su recolección de alimento siguen el mismo camino y refuerzan la feromona sobre este. Si existen varios caminos hacia la fuente de alimento las hormigas seleccionan el camino que va a ser recorrido basándose en la concentración de feromona sobre los caminos existentes [10].

Las hormigas que recorren caminos más cortas hacia las fuentes de alimento, regresan más rápido que aquellas que seleccionaron caminos más largos. De este modo el camino más corto poseerá una mayor concentración de feromona, atrayendo a un número mayor de hormigas [10].

3. PARTE EXPERIMENTAL

Para la comparación estadística entre los algoritmos AG, AM, SI, AS y ACO, se implementaron las pruebas no paramétricas de Friedman, Friedman Alineado y Quade como en [11]. En la experimentación se usó el problema de TSP, simétrico (KroA100, KroA150, KroA200, KroB100, KroB150, KroB200, KroC100, KroD100 y KroE100) y asimétrico (BR17, FTV44, FT53, FT70) obtenidas de Ruprecht Karls Universitat Heidelberg [12], Cada algoritmo fue ejecutado 33 veces para poder aplicar las pruebas estadísticas [13] y en cada ejecución se obtuvo el desempeño de la mejor hormiga o el mejor individuo de la población según sea el caso. De dicho conjunto de ejecuciones se obtuvo la mediana como representante estadístico para aplicar las pruebas no paramétricas. Como condición de paro se utilizó el criterio de 100,000 llamadas a función para cada ejecución.

Se resumen los ajustes y configuraciones de los algoritmos en la siguiente tabla:

	A. Genético	Memético	Sistema Inmune	Ant system	Ant Colony Optimization
Población	33[b]	33[b]	33[b]	-	-
# de hormigas	-	-	-	33[b]	33[b]
Llamadas a función	100,000	100,000	100,000	100,000	100,000
Elitismo %	20	20	20	-	-
% de hipermuta	-	-	30	-	-
# de hijos	-	-	4	-	-
K-flip	-	1	-	-	-
Op. cruza	(OX)	(OX)	(OX)	-	-
Op. Selección	Ruleta pesada	Ruleta pesada	Ruleta pesada	-	-
Op. Muta	Swap-two	Swap-two	Swap-two	-	-
Feromona Ini. (Tau)	-	-	-	0.3[a]	0.3[a]
I. Feromona (Alpha)	-	-	-	0.7[a]	0.7[a]
I. Distancia (Beta)	-	-	-	0.7[a]	0.7[a]
I. Evaporación (Rho)	-	-	-	0.8[a]	0.8[a]
% Explotación (q_0)	-	-	-	-	0.9[a]

a= Dorigo M. Ant colonies for the travelling salesman problem [1996] b= Grinstead C. Central Limit Theorem) [1997]

3.1 Resultados

Se realizaron experimentos para resolver el TSP en 13 instancias de prueba en un enfoque de minimización para encontrar la ruta de menor distancia entre las ciudades de cada instancia dada. Para cada instancia se realizaron 33 experimentos para obtener la ruta mínima de cada experimento, para después obtener la mediana de los 33 experimentos para emplearla como representante en las pruebas estadísticas no paramétricas.

En la tabla siguiente se muestra una selección de las instancias utilizadas durante los experimentos, para el caso simétrico (KroA100, KroB100, KroC100) y el asimétrico (FTV44, FT53, FT70, BR17), donde se muestra la mediana de las distancias obtenidas por cada algoritmo para cada instancia:

Instancia TSP	Optimo Conocido	Ant Colony Optimization	Ant system	Algoritmo Memético	Algoritmo Genético	Sistema Inmune
KROA100	21282	21393	31150	43287	71731	73336
KROB100	22141	23554	32068	42174	68581	73225
KROC100	20749	22005	30543	40224	69391	72756
FTV44	1613	17354	19676	2168	2612	2700
FT53	6905	7953	8938	10247	12898	12129
FT70	38673	39371	41763	43125	51840	50452
BR17	39	39	39	40	41	39

Declaración de hipótesis:

H_0: No existe evidencia sobre la diferencia en el desempeño de los algoritmos.
H_1: Existe evidencia sobre las diferencias en el desempeño de los algoritmos.

Pruebas Ómnibus con nivel de significancia α= 0.05 [11]

F_F	7.714285714	F_{AR}	10.00192493	F_Q	8.59057072
P-valor	0.02112828	P-valor	0.006731465	P-valor	0.004835861
Resultado	Se rechaza H_0	Resultado	Se rechaza H_0	Resultado	Se rechaza H_0

Prueba post-hoc (prueba a pares) de F_{AR} con nivel de significancia α = 0.05 [11]

Algoritmo de control	Memético		Algoritmo de control	Ant Colony Optimization	
	Z	P-valor		Z	P-valor
AM vs AG	5	1.14661E-06	ACO vs AS	2.121320344	0.033894854
AM vs SI	5.285714286	2.5043E-07	ACO vs AM	4.242640687	2.20905E-05

Se acepta H_1: Existe evidencia sobre las diferencias en el desempeño de los algoritmos. [11]

De acuerdo a los resultados de las tablas se puede observar evidencia de que el ACO presenta evidencia suficiente para indicar que hay diferencias de rendimientos con respecto a los algoritmos AG, AM, SI y AS.

4. CONCLUSIONES

Durante la experimentación se observó que el resultado de las pruebas ómnibus Friedman, Friedman de rangos alineados y Quade, concuerdan en los resultados obtenidos, en que hay diferencias de rendimiento entre los algoritmos, al realizar pruebas Post-Hoc las mismas pruebas muestran que de entre los algoritmos evolutivos, el AM es el que presenta las diferencias con respecto a SI y AG. Y que el ACO a su vez supera a AM y AS.

El ACO en cada una de las instancias del problema TSP logró obtener resultados muy cercanos al óptimo conocido por encima de los demás algoritmos, esto sustentado por las pruebas estadísticas no paramétricas. Con las configuraciones empleadas en cada algoritmo para este experimento se evidencia su diferencia de rendimiento con respecto a las demás metaheurísticas. El ACO al ser un algoritmo hecho con el propósito de resolver el TSP, muestra las restricciones específicas del problema dentro de la metaheurística, haciéndolo un método ad-hoc para resolver problemas del modelo TSP. Esto propicia que alcance resultados más próximos a la respuesta optima conocida de cada instancia del problema.

Bajo las condiciones y configuraciones dadas estos fueron los resultados obtenidos, sin embargo no existe la certeza de que estos siempre sean posibles. En los métodos aproximados unas de las variables de mayor importancia, es si hay tiempo suficiente para dar con una respuesta y si la calidad de la misma es una característica restrictiva importante. Restricciones del problema como estas, afectan los resultados de la respuesta obtenida. Por lo anterior el diseño ad-hoc que le permitió lograr estos rendimientos al ACO, es el mismo que podrían evitar aproximar resultados con un buen rendimiento para condiciones y/o características del problema diferentes. El desarrollo de una aproximación que permita lidiar con estos cambios, y que se adapte a las condiciones del problema es el objetivo siguiente de nuestra investigación.

BIBLIOGRAFÍA

1. Alonso, S., Cordon, O., Fernandez, I., Herrera, F. (2001). La metaheurísticas de optimización basada en colonias de hormigas: modelos y nuevos enfoques. Optimización inteligente: técnicas de inteligencia computacional para optimización, 261-314.

2. Torn, A. and A. Zilinskas. 1989.”Global Optimization”. Berlin: Springer. Vaidyanathan, P. P. (1993). Multirate Systems and Filter Banks. Englewood Cliffs, New Jersey: Prentice Hall. Valenzuela-Rendon, M. and E. Uresti.
3. Rafael Martí, Gerhard Reinelt. 2011.”The Linear Ordering Problem”, Exact and Heuristic Methods in Combinatorial Otimization. Aplied Mathematical Science 175. Springer Verlag Berlin Heidelber. Page 17.
4. S. Lin and B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, Oper. Res. 21 (1973), pp. 498–516.
5. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, The Traveling Salesman Problem: G.E. Re Guided Tour of Combinatorial Optimization, Wiley and Sons, New York (1985).
6. M. Padberg and G. Rinaldi, Optimization of a 532-city symmetric traveling salesman problem by branch and cut, Oper. Res. 6 (1987), pp. 1–7.
7. AGUST´IN, B. (1998). Aplicación de algoritmos gen éticos al diseño ´óptimo de sistemas de distribución de energía eléctrica. Zaragoza.
8. Moscato, P., Cotta, C. (2003). Una Introducción a los Algoritmos Meméticos. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 7(19), 131-148.
9. J.D. Schaffer and L.J. Eshelman. .On crossover as an evolutionary viable strategy”. In R.K. Belew and L.B. Booker, editors. Proceedings of the 4th International Conference on Genetic Algorithms, pages 61-68, Morgan Kaufmann, 1991.
10. Dorigo, M.; Maniezzo, V.; Colorni, A., .Ant system: optimization by a colony of cooperating agents, ”Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.26, no.1, pp.29,41, Feb 1996
11. Derrac, J., Garc´ıa, S., Molina, D., Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), 3-18.
12. Ruprecht Karls Universitat Heidelberg. Extraido de: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95
13. Grinstead, Charles M.; Snell, J. Laurie (1997). ((9. Central Limit Theorem)). Introduction to Probability (2 edición). AMS Bookstore. pp. 325–360. ISBN 0821807498.

Agradecimientos.

Al Consejo Nacional de la Ciencia y la Tecnología (CONACyT) por brindar los recursos económicos para permitir elaborar este trabajo y a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de León por brindar las facilidades para cursar la Maestría en Ciencias en Ciencias de la Computación. (CVU/Becario): **595547/309062**